BAB II
TEORI PENUNJANG

2.1 Laptop

Laptop merupakan sebuah *computer portable*(yang bisa dibawa kemana-mana) dan terintegrasi pada casing. Menurut Robbert H. Blissmer(1985), “laptop di definisikan sebagai alat electronic yang dapat melakukan serangkaian tugas yaitu menerima input, memproses input sesuai dengan intruksi yang diberikan menyimpan perintah dan hasil pengolahannya serta menyediakan output dalam bentuk informasi"[12]. Laptop juga kadang-kadang disebut sebagai notebook. Tetapi diluar sana, laptop ini dikenal juga dengan istilah *notebook* atau *powerbook.* Secara umum, laptop ini memiliki berat berkisar mulai dari satu hingga enam kilogram, tergantung dari ukuran, bahan serta spesifikasi laptop tersebut.

Laptop dirancang agar bisa digunakan secara *mobile* dengan ukuran yang kecil dan cukup ringan untuk diletakan dipangkuan seseorang ketika digunakan. Sebuah laptop telah terintegrasi dengan sebagian besar komponen khas yaitu seperti halnya *computer desktop*, contohnya layar *monitor*, *keyboard*, perangkat penunjuk yang dikenal juga *trackpad*, speaker dan baterai yang telah disusun pada satu unit kecil yang ringan. Baterai yang bisa di charge seperti laptop ini biasanya menyimpan *energy* untuk digunakan selama dua sampai tiga jam dalam keadaan awal, hal ini tergantung pada konfigurasi dan manajemen daya laptop itu sendiri.[11]

2.2 Android

Android adalah *sistem operasi* atau OS berbasis linux yang diperuntukan khusus untuk mobile device seperti *smartphone* atau *PC tablet*, persis seperti symbian yang dipergunakan oleh Nokia dan BlackBerry OS, jelasnya seperti microsoft windows yang sangat dikenal baik oleh para pengguna komputer dan laptop, jika kita analogikan, *Andoid* adalah windows nya sedangkan *smartphone* atau *handphone* atau *tablet* adalah unit komputernya.[13]



Gambar 2.1 Lambang android

Android adalah *sistem operasi* atau OS berbasis linux yang diperuntukan khusus untuk mobile device seperti *smartphone* atau *PC table*, persis seperti symbian yang dipergunakan oleh Nokia dan BlackBerry OS, jelasnya seperti microsoft windows yang sangat dikenal baik oleh para pengguna komputer dan laptop, jika kita analogikan, *Andoid* adalah windows nya sedangkan *smartphone* atau *handphone* atau *tablet* adalah unit komputernya.[13]

Android di ambil dari nama perusahaan penemunya yaitu Android.inc yang kemudian di akuisisi oleh Google pada pertengahan tahun 2005 dan mengubah nama penyedia aplikasi Android dari Android market menjadi *Google play*. Dengan sistem distribusi *open sources* yang di gunakan memungkinkan para pengembang untuk menciptakan beragam apikasi menarik yang dapat dinikmati oleh para penggunanya, seperti game, *chatting* dan lain-lain.System operasi *android* tersebut secara resmi dirilis pada tahun 2007, bersamaan dengan didirikannya sebuah perusahan open hanset alliance, konsorsium dari beberapa perusahaan perusahaan perangkat keras, perangkat lunak, serta telekomunikasi yang memiliki tujuan untuk memajukan standar terbuka dari perangkat seluler. Ponsel yang berbasis system operasi android pertama dijual pada bulan oktober 2008.

Sejarah android pada mulanya berasal dari perusahaan bernama Android, Inc. didirikan tempatnya di Palo Alto, California pada Oktober tahun 2003 oleh Andy Rubin (pendiri Danger), Rich Miner seorang pendiri Wildfire Communications, Inc., Nick Sears seorang mantan VP T-Mobile, dan Chris White seorang kepala desain dan pengembangan antarmuka WebTV untuk

mengembangkan sebuah “perangkat seluler pintar yang lebih sadar tentang lokasi dan preferensi penggunanya”. Tujuan awal dari perkembangan tersebut pada mulanya diperuntukan bagi camera digital, namun disadari bahwa pasar dari kamera digital tidak besar potensinya, dan pengembangan android lalu dialihkan pada pasar telepon pintar atau smartphone untuk menyaingi Symbian serta Windows Mobile (iPhone Apple pada saat itu belum dirilis).

Meskipun para pengembang android tersebut merupakan pakar-pakar teknologi yang berpengalaman, Android Inc. dijalankan secara diam-diam dan hanya diungkapkan bahwa para pengembang tersebut sedang berusaha menciptakan sebuah perangkat lunak yang masih diperuntukan untuk telepon seluler. Masih pada tahun yang sama, Andy rubin kehabisan uang, Steve Perlman adalah seorang teman dekat Andy Rubin dan meminjaminya $10.000 tunai serta menolak tawaran saham di perusahaan.

Sejak tahun 2008 android mulai secara bertahap melakukan sejumlah pembaruan atau update untuk meningkatkan kinerja dari system operasi tersebut dengan menambahkan fitur baru, memperbaiki bug pada versi android yang sebelumnya.

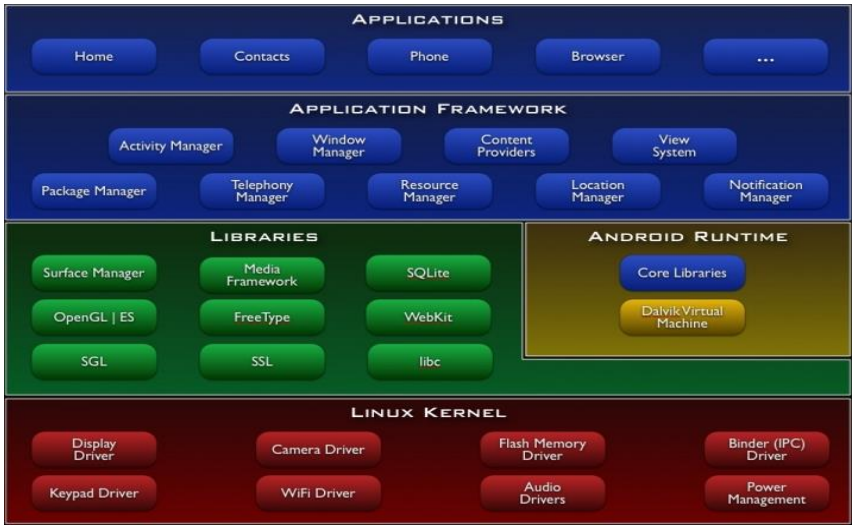


Gambar 2.2 Perkembangan Sistem Operasi Android

Setiap versi yang dirilis dinamakan secara alfabet dengan berdasarkan nama sebuah makanan pencuci mulut, seperti *cup cake, donut* dan sebagainya.

2.2.1 Arsitektur android

Secara umum arsitektur android di bagi 5 *layer* elemen, setiap layer terdiri dari beberapa program yang mendukung fungsi-fungsi spesifik dari sistem operasi. Secara garis besar arsitektur *android* dapat dijelaskan dan digambarkan pada Gambar 2.3:



Gambar 2.3 Arsitektur Android

1. *Application Layer*

Aplication Layer merupakan lapisan paling tampak pada pengguna ketika digunakan tanpa mengetahui proses yang terjadi dibalik lapisan aplikasi. Lapisan ini berjalan dalam android runtime dengan menggunakan kelas dan service yang tersedia pada framework aplikasi.

2. *Aplication Framework Layer*

Merupakan kerangka aplikasi yang menyediakan kelas-kelas yang dapat digunakan untuk mengembangkan aplikasi android. Selain itu, juga menyediakan abstraksi generic untuk mengakses perangkat, serta mengatur tampilan user interface dan sumber daya aplikasi.

3. *Libraries Layer*

Merupakan layer yang didalamnya terdapat fitur-fitur android, biasanya para pembuat aplikasi mengakses libraries untuk menjalankan aplikasinya

berjalan di atas karnel. *Layer* ini meliputi berbagai *library* C/C++ inti seperti Libc dan SSL, serta:

1. *Libraries* media untuk pemutaran media audio dan video.
2. *Libraries* untuk manajemen tampilan.
3. *Libraries Graphics* mencakup SGL dan OpenGL untuk grafis 2D dan 3D.
4. *Libraries* SQLite untuk mendukung *database.*
5. *Libraries* SSL dan WebKit terintegrasi dengan *web browser* dan *security*.
6. *Libraries* Live Webcore mencakup modern *web browser* dan *engine embeded web view*.
7. *Libraries* 3D yang mencakup implementasi OpenGL ES 1.0 API's.

4. *Android Run Time Layer*

Layer ini merupakan*layer* yang membuat aplikasi android dapat dijalankan dimana dalam prosesnya menggunakan implementasi Linux. *Dalvix Virtual Machine*(DVM) merupakan mesin yang membentuk dasar kerangka aplikasi android. Di dalam android *run time* dibagi menjadi dua bagian, yaitu:

a. *Core Libraries*

Aplikasi android dibangun dalam bahasa java, sementara *Dalvik* sebagai virtual mesinnya bukan *virtual machine* java, sehingga diperlukan sebuah libraries yang berfungsi untuk menterjemahkan bahasa java/C yang ditangani oleh *Core Libraries.*

b. *Dolvik Virtual Machine*

Virtual mesin berbasis register yang di optimalkan untuk menjalankan fungsi-fungsi secara efisien, dimana merupakan pengembangan yang mampu membuat linux karnel untuk melakukan threading dan manajemen tingkat rendah.

5. *Linux Kernel Layer*

Merupakan *layer* dimana inti dari sistem operasi android itu berada. *Layer*ini berisi *file-file system* yang mengatur sistem *processing, memory, resource, drivers,* dan sistem-sistem operasi android lainnya.[2]

2.3 Augmented Reality

Augmented Reality (AR) adalah sebuah istilah untuk lingkungan yang menggabungkan dunia nyata dan dunia *virtual* yang dibuat oleh komputer sehingga batas antara keduanya menjadi sangat tipis. Menurut Ronald Azuma(1997) mendefinisikan *Augmented Reality* sebagai sistem yang memiliki karakteristik sebagai berikut:

1. Menggabungkan lingkungan nyata dan virtual
2. Berjalan secara interaktif dalam waktu nyata
3. Integrasi dalam tiga dimensi (3D)

Secara sederhana AR dapat didenisikan sebagai lingkungan nyata yang ditambahkan objek virtual. Penggabungan objek nyata dan virtual dimungkinkan dengan teknologi *display* yang sesuai, interaktivitas dimungkinkan melalui perangkat-perangkat input tertentu.

AR merupakan variasi dari *Virtual Environments* (VE), atau yang lebih dikenal dengan istilah *Virtual Reality* (VR). Teknologi VE membuat pengguna tergabung dalam sebuah lingkungan virtual secara keseluruhan. Ketika tergabung dalam lingkungan tersebut, pengguna tidak dapat melihat lingkungan nyata di sekitarnya. Sebaliknya, AR memungkinkan pengguna untuk melihat lingkungan nyata, dengan objek virtual yang ditambahkan atau tergabung dengan lingkungan nyata. Tidak seperti VR yang sepenuhnya menggantikan lingkungan nyata, AR sekedar menambahkan atau melengkapi lingkungan nyata.[1]

Tujuan utama dari AR adalah untuk menciptakan lingkungan baru dengan menggabungkan interaktivitas lingkungan nyata dan virtual sehingga pengguna merasa bahwa lingkungan yang diciptakan adalah nyata. Dengan kata lain, pengguna merasa tidak ada perbedaan yang dirasakan antara AR dengan apa yang mereka lihat/rasakan di lingkungan nyata. Dengan bantuan teknologi AR (seperti visi komputasi dan pengenalan objek) lingkungan nyata disekitar kita akan dapat berinteraksi dalam bentuk digital (*virtual*). Informasi tentang objek dan lingkungan disekitar kita dapat ditambahkan kedalam sistem AR yang kemudian

informasi tersebut ditampilkan diatas layer dunia nyata secara realtime seolah-olah informasi tersebut adalah nyata. Informasi yang ditampilkan oleh objek virtual membantu pengguna melaksanakan kegiatan-kegiatan dalam dunia nyata. AR banyak digunakan dalam bidang-bidang seperti kesehatan, militer, industri manufaktur dan juga telah diaplikasikan dalam perangkat-perangkat yang digunakan orang banyak, seperti pada telepon genggam.[1]



Gambar 2.4 Contoh Augmented Reality Sederhana

Berikut merupakan contoh dari penerapan teknologi *Augmented Reality* sederhana dapat dilihat pada Gambar 2.4.

Augmented Reality dapat di klasifikasikan menjadi dua berdasarkan ada tidaknya penggunaan *marker* yaitu:

1. *MarkerAugmented Reality*

   Sebuah metode yang memanfaatkan marker yang biasanya berupa ilustrasi hitam dan putih berbentuk persegi atau lainnya dengan batas hitam tebal dan latar belakang putih. Melalui posisi yang dihadapkan pada sebuah kamera komputer atau *smartphone*, maka komputer atau *smartphone* tersebut akan melakukan proses menciptakan dunia virtual 2D atau 3D. Marker Based Tracking ini sudah lama dikembangkan sejak 1980 dan pada awal 1990 mulai dikembangkan untuk pengguna *Augmented Reality.*

2. *Markerless Augmented Reality*

Salah satu metode Augmented Reality yang saat ini sedang berkembang adalah metode *Markerless Augmented Reality*. Dengan menggunakan metode ini maka pengguna tidak perlu menggunakan sebuah marker untuk menampilkan objek 3D atau yang lainnya. Sekalipun dengan markerless namun aplikasi tetap berjalan dengan melakukan pemindaian terhadap objek, namun ruang lingkup yang dipindai lebih luas dibanding dengan *Marker Based Tracking*.

Saat ini Augmented Reality dikembangkan oleh perusahaan terbesar di dunia yaitu Total Imersion, yang telah membuat berbagai macam teknik Markerless Tracking sebagai teknologi andalan perusahaan, seperti *Image Target, Face Tracking, 3D Object Tracking, dan Motion Tracking.*

a. *Image Target*

Image Target adalah sebuah metode dimana marker yang dibaca oleh sistem AR adalah sebuah Gambar, baik Gambar hitam putih atau Gambar berwarna.

b. *Face Tracking*

Dengan menggunakan algoritma yang telah dikembangkan, komputer dapat mengenali wajah manusia secara umum dengan cara mengenali semua komponen wajah manusia kemudian akan mengabaikan objek-objek lain di sekitarnya seperti pohon, rumah, dan benda-benda lainnya.

c. *3D Object Tracking*

Berbeda dengan *Face tracking* yang hanya mengenali wajah manusia secara umum,teknik *3D Object Tracking* dapat mengenali semua bentuk benda yang ada disekitar, seperti mobil, meja, televisi, dan lain sebagainya.

d. *Motion Tracking*

Teknik ini merupakan teknik yang membuat komputer dapat menangkap gerakan. *Motion Tracking* telah mulai digunakan secara ekstensif untuk memproduksi film-film yang mencoba mensimulasikan gerakan. Contohnya pada film Avatar, dimana James Cameron menggunakan teknik ini untuk membuat film tersebut dangn menggunakannya secara *real-time.*[1]

2.4 *Tools* pendukung pembuatan aplikasi

Berikut dibawah ini ada beberapa tools (pembantu) yang mendukung untuk pembuatan aplikasi media promosi ini sebagai berikut :

2.4.1 *Tools* yang mendukungJava Development Kit (JDK)

JDK adalah *software development* kit merupakan seperangkat kode, *library*, form dan paket yang berguna untuk membuat aplikasi atau applet java. Di dalam JDK terdapat Java Runtime Environment (JRE) yang akan memproses byte code yang dihasilkan dari interpenter java. Tugas dari interpenter java ini adalah untuk mengcompile source.java menjadi source. Class yang selanjutnya source.class inilah yang akan dijalankan oleh JRE.

2.4.2 Software Development Kit (SDK)

SDK adalah suatu kumpulan dari library dan toolkit seperti emulator untuk mengembangkan atau menciptakan aplikasi untuk suatu perangkat lunak, perangkat *computer*, system operasi atau platform. Di dalam SDK terdapat tools yang dibutuhkan dalam pengembangan android, diantaranya yaitu adb shell yang merupakan singkatan dari Android Development Bridge yang dapat menjalankan terminal android seperti terminal pada system operasi linux, dan command yang terdapat dalam adb shell sendiri seperti command linux pada umumnya.

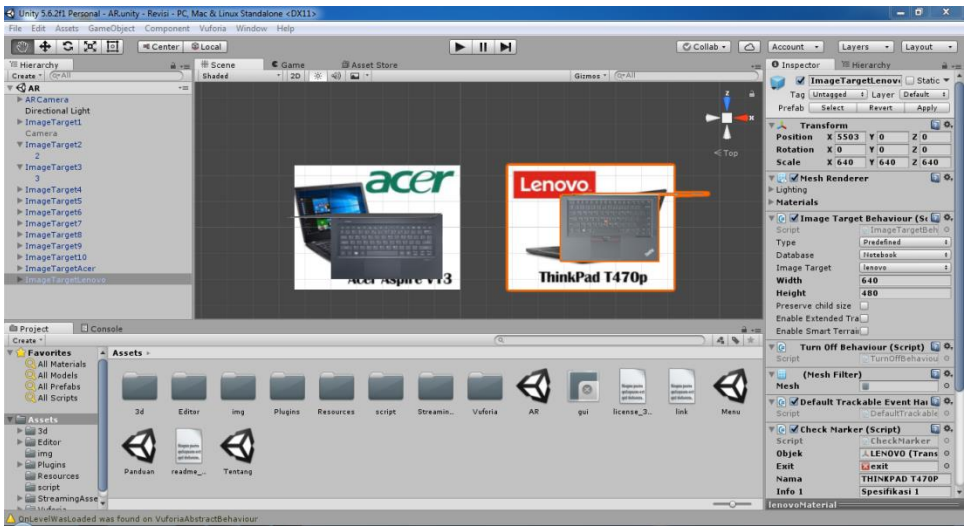
2.4.3 Unity 3D Engine

Unity Engine merupakan suatu software *game engine* yang terus berkembang saat ini. Penggunaan *engine* versi *free* masih dibatasi dengan beberapa fitur yang dikurangi atau bonus *modul/prefab* tertentu yang ditiadakan dan hanya tersedia untuk pengguna berbayar. *Unity Engine* dapat mengolah beberapa data seperti gambar tiga dimensi, suara, tekstur, dan lain sebagainya.

Keunggulan dari *unity engine* ini yaitu, dapat menangani grafik dua dimensi dan tiga dimensi. Namun *engine* ini lebih konsentrasi pada pembuatan grafik tiga dimensi. Dari beberapa *game engine* yang sama-sama menangani grafik tiga dimensi, *Unity Engine* dapat mengani lebih banyak.

Beberapa diantaranya yaitu Windows, MacOS X, iOS, PS3, wii, Xbox 360, dan android yang lebih banyak daripada *game engine* seperti *Source Engine*, *Game Marker*, *Unigine*, *id Tech 3 Engine*, *Blender Game Engine*, *NeoEngine*, *Quake Engine*, *C4 Engine* atau *game engine* lain. *Unity Engine* memiliki kerangka kerja (*framework*) lengkap utnuk pengembangan professional. Sistem inti *engine* ini menggunakan beberapa pilihan bahasa pemrograman, diantaranya C#, javascrit maupun boo.[6]

Unity3D editor menyediakan beberapa alat untuk memepermudah pengembangan yaitu *Unity Tree* dan *Terrain Creator* atau mempermudah pembuatan *vegetasi* dan *terrain* serta *MonoDevelop* untuk proses pemrograman. Tampilan dari software Unity3D *engine* dapat dilihat pada Gambar 2.5.



Gambar 2.5Tampilan Unity3d

2.4.4 Vuforia Augmented Reality

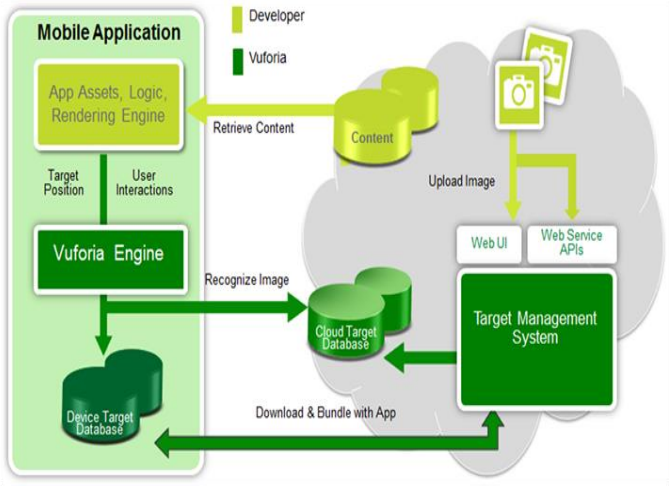
Vuforia merupakan *librarry* untuk *Augmented Reality* yang dikembangkan oleh Qualcomm, yang menggunakan sumber konsisten mengenai *computer vision* yang fokus pada *image recognition*. Vuforia mempunyai banyak fitur-fitur dan

kemampuan, yang dapat membantu pengembangan untuk mewujudkan pemikiran mereka tanpa adanya batas secara teknikal.

Dengan *support* untuk sistem operas iOS, android, dan Unity3D, *platform* Vuforia mendukung para pengembang untuk membuat aplikasi yang dapat digunakan di hampir seluruh jenis *smartphone* dan *tablet*.

Dalam pengembangannya juga pengguna diberikan kebebasan untuk mendesain dan membuat aplikasi yang mempunyai kemampuan antara lain:

1. Teknologi *computer vision* tingkat tinggi yang mengijinkan *developer* untuk membuat efek khusus pada *mobile device*.
2. Terus-menerus mengenali *multiple image*.
3. *Tracking* dan detection tingkat lanjut.
4. Solusi pengaturan *database* gambar yang fleksibel.



Gambar 2.6 Struktur Vuforia

Target pada Vuforia merupakan obyek pada dunia nyata yang dapat dideteksi oleh kamera, untuk menampilkan obyek virtual. Beberapa jenis target pada Vuforia adalah:

1. *Image Targets*, contoh: foto, papan permainan, halaman majalah, sampul buku, kemasan produk, poster, kartu ucapan.
2. *Frame markers*, tipe *frame*gambar 2D dengan patteren khusus yang dapat digunakan sebagai permainan.
3. *Multi-target*, contohnya kemasan produk atau produk yang berbentuk kotak ataupun persegi. Jenis ini dapat menampilkan gambar sederhana *Augmented Reality* 3D.
4. *Virtual buttons,* yang dapat membuat tombol sebagai daerah kotak sebagai sasaran Gambar.

2.4.5 Arsitektur Vuforia

Vuforia SDK memerlukan beberapa komponen penting agar dapat bekerja dengan baik. Komponen-komponen tersebut antara lain:

a. Kamera

Kamera dibutuhkan untuk memastikan bahwa setiap *frame* ditangkap dan diteruskan secara efisian ke *tracker*. Para developer hanya tinggal memeberi tahu kamera kapan mereka mulai menangkap dan berhenti.

b. Image Converter

Mengkonfersi format kamera (misalnya YUV12 kedalam format yang dapat dideteksi oleh OpenGL (misalnya RGB565) untuk *tracking* (misalnya *luminance*).

c. Tracker

Mengandung algoritma *computer vision* untuk melakukan pelacakan objek nyata yang diambil dari kamera. Objek tersebut dievaluasi dan hasilnya akan disimpan dalam *state object*yang akan digunakan oleh video *background renderer* dan dapat dikases dari *application code*.

d. Video Background Render

Me-*render*gambar dari kamera yang tersimpan di dalam *state object*. Performa dari *video background render* sangat bergantung pada *device* yang digunakan.

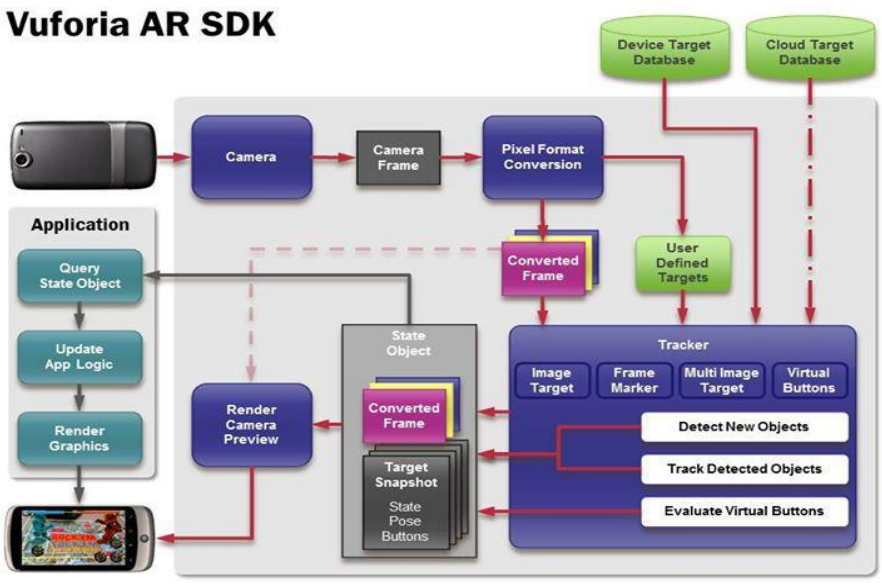
e. *Application Code*

Melibatkan analisasi dari semua komponen diatas. Selama objek yang dikehendaki diubah prosesnya, maka *application code* harus diubah berdasarkan lokasi objek virtual.

f. *Target Resource*

Dibuat menggunakan online *Target Management System*. Assets di unduh berisi sebuah konfigurasi xml config.xml yang memungkinkan developer untuk mengkonfigurasi beberapa fitur dalam trackable dan binary file yang berisi database trackable.

Berikut adalah gambaran dari diagaram alur data Vuforia, dapat dilihat pada Gambar 2.7.



Gambar 2.7 Diagram Alir Data Vuforia

2.4.6 Blender

Blender 3D merupakan aplikasi untuk membuat gambar 3D yang dapat digunakan oleh siapapun (*Opensource*). tidak hanya membuat gambar 3D, Blender dapat digunakan untuk pengeditan video, membuat animasi, bahkan membuat game .

Terdapat beberapa fasilitas modeling yang dapat membantu dalam pembuatan gambar 3D pada Blender diantaranya:

1. Mesh Modeling

   Teknik dasar modeling yang digunakan dalam pembuatan gambar 3D di sini terdapat teknik box modeling yang terdiri dari object primitif (box, sphare, cylinder dan sebagainya, Objek tersebut biasanya digunakan sebagai dasar dalam pembuatan gambar 3D. Blender mengenali face dengan 4 vertek atau 4 sisi sehingga pembuatan gambar akan lebih mudah.
2. Metaball

   Metaball merupakan fasilitas modeling yang jarang dipakai, metaball merupakan sebuah object berbentuk bola dan bersifat seperti cairan atau tanah liat, terdapat 2 buah metaball yaitu metaball positif dan negatif jika metaball positif didekatkan dengan metaball positif akan saling menyatu satu sama lain, namun penggunaanya pada Blender masih belum sempurna, tools masih terbatas.
3. Curve, NURBS (Non-Uniform Rational B-Slines), dan Survace

   Ini merupakan fasilitas umum, terdapat pula pada software Blender 3D, kelebihan menggunan pemodelan ini, data yang tersimpan lebih sedikit, hasil lebih bagus, namun terdapat kekurangan pada model ini, sangat sulitnya menambahkan detail yang kompleks, karena jumlah poligonnya sangat sulit dikendalikan[11].

2.5 Pemograman Berorientasi Objek

Metode berorientasi objek adalah suatu strategi pembangunan perangkat lunak yang mengorganisasikan perangkat lunak sebagai kumpulan objek yang berisi data dan operasi yang diberlakukan terhadapnya. Metode berorientasikan objek merupakan suatu cara bagaimana sistem perangkat lunak dibangun melalui pendekatan objek secara sistematis. Metode berorientasi objek didasarkan pada

penerapan prinsip-prinsip pengolahan kompleksitas. Metode berorientasi objek meliputi rangkaian aktivitas, analisis berorientasi objek, perancangan berorientasi objek, pemrograman berorientasi objek, dan pengujian berorientasi objek. [8]

Berikut beberapa contoh bahasa pemrograman yang mendukung pemrograman berorientasi objek:

1. Bahasa pemrograman Smalltalk
2. Bahasa pemrograman Eiffel
3. Bahasa pemrograman C++
4. Bahasa pemrograman (web) PHP
5. Bahasa pemrograman Java

2.5.1 Unifield Modelling Language (UML)

UML merupakan bahasa yang telah menjadi standar untuk melakukan perancang, visualisasi dan mendokumentasikan sistem pada piranti lunak, dengan UML kita dapat memodelkan aplikasi yang dapat berjalan pada sistem operasi dan dapat ditulis menggunakan berbagai macam bahasa pemrograman, namun bahasa pemrograman yang sering digunakan dalam pembuatan *software* yaitu bahasa yang berorientasi objek seperti C++, JAVA, C# atau VB.NET [14].

Notasi UML merupakan hasil pengembangan 3 notasi yang telah ada sebelumnya, yaitu : [16]

- *Grady Booch* OOD (*Object-Oriented Design*).

- *Jim Rumbaugh* OMT (*Object Modeling Technique*).

- *Ivar Jacobson* OOSE (*Object-Oriented Software Engineering*).

Beberapa konsep dasar UML, yaitu *structural*, *classification*, *dynamic*, *behavior*, dan model *management*. Dapat kita lihat pada gambar di bawah ini, sama halnya dengan apa yang ada pada gambar UML mendefinisikan beberapa diagram: [14]

1. *Use Case* Diagram

Menekankan kepada apa yang akan diperbuat sistem, sebuah use case menggambarkan interaksi aktor dengan sistem, aktor dapat berupa manusia maupun mesin. Sebuah *use case* dapat *include* terhadap *use case* lain, dapat digambarkan ketika ada *use case* yang *include* dengan *use case* yang di eksekusi maka use case yang *include* tersebut akan ikut dieksekusi, tidak hanya satu use case yang dapat *include* pada sebuah use case, lebih dari satu *use case* dapat *include* pada sebuah use case. Sebuah *use case* pun dapat meng-*extend use case* lain dengan *behavior*-nya sendiri.

2. *Class* Diagram

   Class Diagram adalah sebuah sepesifikasi yang diinstansikan akan menghasilkan objek, *class* diagram memiliki tiga area pokok : nama, atribut, dan metode. Pada atribut dan metode dapat memiliki salah satu sifat ini. *private, protected, public* . Hubungan antara *class* pada *class* diagram:

   a. Asosisasi, yaitu hubungan statis antara class. Asosiasi *class* pun memiliki atribut berupa class lain, panah penunjuk menunjukan arah *query* antara *class*.

   b. Agregasi, yaitu hubungan yang menyatakan bagian yang di dalamnya terdiri atas *class*-*class*.

   c. Pewarisan, yaitu hubungan hirarki antara *class*. *Class* dapat diturunkan kepada *class* lain.

   d. Hubungan dinamis, yaitu rangkaian pesan yang di-*passing* dalam satu *class* kepada *class* lain.

3. *Statechart* Diagram

   Diagram ini digunakan untuk menggambarkan perpindahan atau perubahan keadaan dari satu *state* ke *state* lainya. Dalam satu *class* dapat memiliki lebih dari satu *state*.

4. *Activity* Diagram

   Diagram ini menggambarkan bagaimana masing-masing alir berjalan. Sehingga *activity* diagram dapat menggambaran proses pararel yang mungkin terjadi pada eksekusi-eksekusi lain.

5. *Sequence* Diagram

   Diagram ini menggambarkan interaksi antara objek di dalam dan objek di sekitar sistem. Diagram ini dapat digunakan untuk menggambarkan skenario dan langkah-langkah yang dilakukan.

6. *Collaboration* Diagram

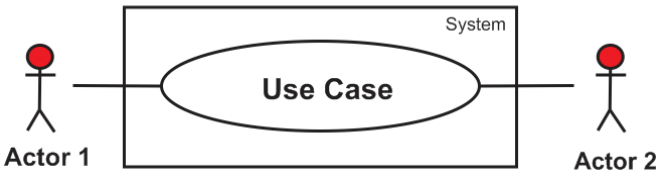
   Diagram ini menggambarkan interaksi antara objek, namun lebih menekankan pada peranan masing-masing objek seperti *sequence* diagram dan bukan merupakan penyampaian pesan, setiap pesan memiliki *sequence number*.

7. *Component* Diagram

   Diagram ini menggambarkan struktur dan hubungan antara komponen perangkat lunak, dengan ketergantungan diantaranya, komponen bisa berupa *interface*, yaitu kumpulan layanan yang disediakan sebuah komponen untuk komponen lain.

2.5.2 Use Case Diagram

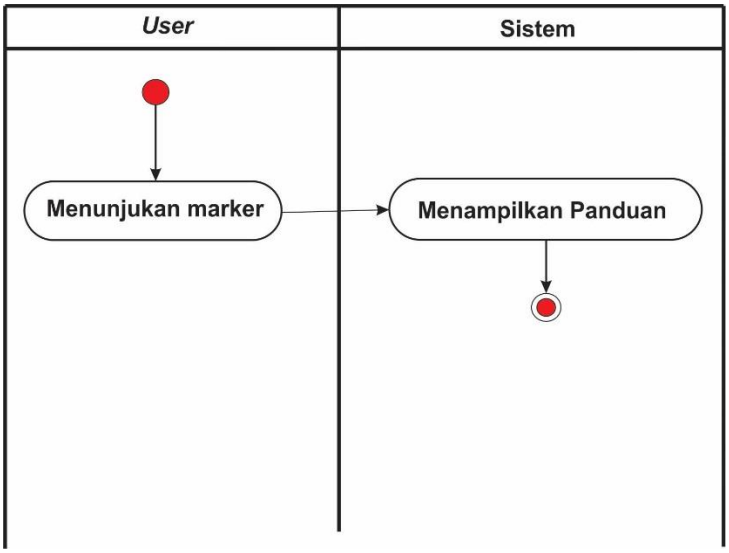
Use Case Diagram merupakan diagram yang dirancang untuk menunjukan fungsionalitas suatu sistem yang berinteraksi dengan dunia luar. *Use case diagram* juga digunakan untuk menjelaskan manfaat suatu sistem jika dilihat dari sudut pandang orang yang berada di luar sistem. Berikut merupakan contoh gambar *Use Case* diagram dapat dilihat pada Gambar 2.8.



Gambar 2.8 Contoh Use Case Diagram

2.5.3 *Activity* Diagram

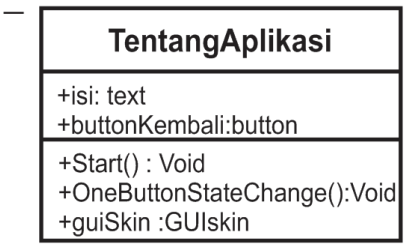
Diagram yang menggambarkan rangkaian aliran dari aktivitas, digunakan untuk mendeskripsikan aktivitas yang dibentuk dalam suatu operasi sehingga dapat digunakan untuk aktivitas lainnya seperti *Use Case*. Berikut merupakan contoh *activity* diagram dapat dilihat pada Gambar 2.9.



Gambar 2.9 Contoh Gambar Activity Diagram

2.5.4 *Class* Diagram

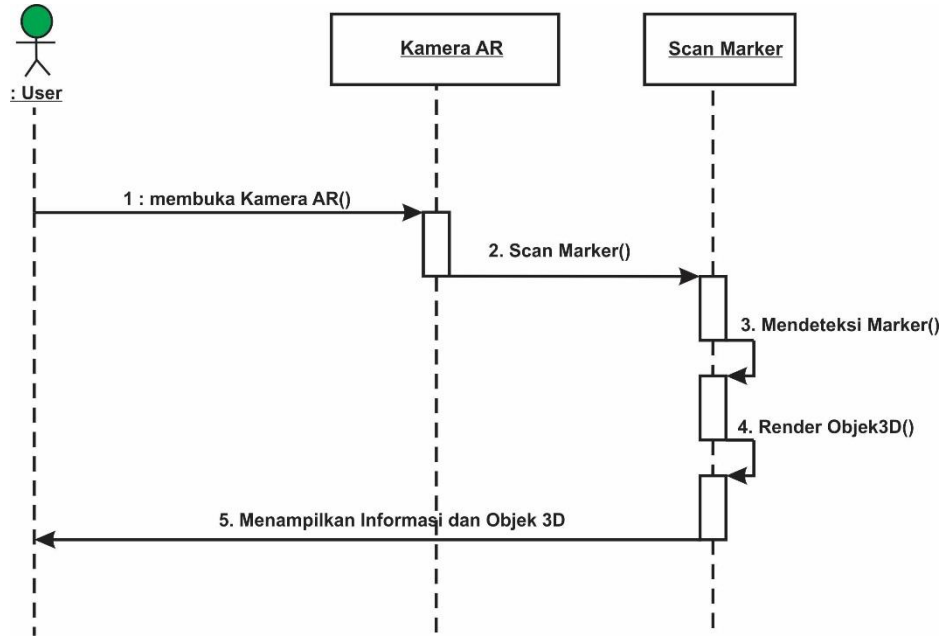
Diagram yang menjelaskan dalam visualisasi struktur kelas-kelas dari suatu sistem. *Class* diagram menunjukan hubungan antar kelas dan penjelasan *detail* tiap-tiap kelas dalam model desain suatu sistem. Berikut merupakan contoh *class* diagram dapat dilihat pada Gambar 2.10.



Gambar 2.10 Contoh Class Diagram

2.5.5 *Squence* Diagram

Menjelaskan interaksi objek yang disusun dalam suatu urutan waktu. Diagram ini secara khusus bersosialisasi dengan *use case*. *Sequence diagram*, memperlihatkan tahap demi tahap yang seharusnya terjadi untuk menghasilkan sesuatu dalam *use case*. Berikut merupakan contoh gambar squence diagram dapat dilihat pada Gambar 2.11.



Gambar 2.11 Contoh Squence Diagram